ISSN: 3030-3346

ADVERSARIAL ANALYSIS OF AUTONOMOUS MULTI-AGENT NETWORK DEFENCE

Suhrobjon Bozorov

TUIT named after Muhammad al-Khwarizmi
E-mail: mr.bozorov95@gmail.com,
ORCID 0000-0002-2377-934X

K E Y W O R D S

autonomous network defense, attack surface, threat modelling, adversarial evasion, Dempster–Shafer conflict, reliability discounting, escalation saturation, multi-agent intrusion detection

A B S T R A C T

Research on the security of machine-learning network defense treats the detector as the object under attack. A system that also acts - installing enforcement rules, resetting sessions, quarantining endpoints and revising its own policy from observed outcomes - acquires an attack surface of its own, and that surface is neither enumerated nor measured across most of the literature. This paper analyses an eight-agent detect-and-respond architecture with evidence-theoretic fusion and a reinforcement-learned response policy as a target rather than a defense. Ten threats are enumerated across four surfaces - the evidence path, the coordination bus, the learning path and the enforcement plane - each mapped to an architectural control and to a stated residual risk. Three findings follow: under constrained protocol-valid perturbation the dominant failure mode is displaced rather than removed, reliability discounting bounds agent compromise only asymptotically, and finally, the conflict gate introduces a failure mode of its own: Each control bounds its threat; none eliminates one, and one creates a new one.

Introduction

The security literature on learning-based network defense asks, almost without exception, whether a detector can be evaded, which presumes the system's only output is a label. A system that acts on its own conclusions - installing a drop rule, resetting a session, quarantining an endpoint, then revising its policy according to what followed - produces outputs that alter the network it observes, each of which an adversary may prefer to control. The question is therefore not only whether the detector can be deceived, but which components other than the detector repay attack.

The system examined here is CMAI-DM, an eight-agent architecture whose detection agents emit Dempster–Shafer mass functions rather than scores, whose correlation agent combines them under reliability discounting and suspends enforcement when conflict exceeds a threshold, and whose mitigation agent selects among seven graduated, reversible, time-bounded actions under a policy learned by Proximal Policy Optimisation [21]. Its detection performance is established elsewhere [24]. Three properties make it an unusual target: evidence is heterogeneous, so evasion tuned against one detector family surfaces as disagreement rather than silence; autonomy is conditional, so the gate may be attacked in preference to the detector; and the system learns from its own operation, placing the update path within reach of the network it defends.

Four contributions follow: a catalogue of ten threats across four attack surfaces, each mapped to a control and to a stated residual risk; a characterisation of behaviour under constrained evasion by decision route rather than by accuracy; a derivation of the exposure window for a compromised agent from the reliability-update dynamics; and the identification of escalation saturation as a failure mode the conflict gate introduces rather than mitigates.

Related Works

(cc) BY

Adversarial machine learning against network detectors is surveyed by He, Kim and Asghar [11], while the methodological critiques of Arp et al. [7], Apruzzese et al. [8] and Jacobs et al. [9] show that reported robustness is frequently an artefact of evaluation design. What this work shares is its scope: the object under attack is a model emitting a label.

Reinforcement learning entered the field as a means of turning that label into a decision [12], [19], [22], and Benaddi et al. [23] formulate detection as a stochastic game, though the action space remains classification. Multi-agent formulations distribute detection across cooperating agents [13]–[17], [25], and collaborative ensembles are surveyed in [26]; in each the agents assign labels rather than act. Systems that genuinely act are rare: Foley et al. [18] operate in CybORG [20] on abstract host state and Long and Jinsong [10] enforce through a control plane under a fixed threshold; federated update paths [27] add another reachable surface. Across this literature, and across work on detector architecture and denial-of-service detection [29], [30], the fusion rule, the coordination substrate, the response machinery and the learning path are treated as implementation detail rather than attack surface - the gap addressed here.

Materials and Methods

System under analysis

Defence is distributed across eight agents. A1 assembles flows and emits a feature vector stamped with the first packet's capture time. A2 (anomaly), A3 (supervised) and A4 (behavioural) receive it in parallel under a deadline and return mass functions over {attack, benign, ignorance} rather than scalar scores. A5 discounts each by a reliability coefficient and combines them under Dempster's rule, yielding a belief interval and conflict measure K. A6 converts belief into a bounded risk score weighted by criticality, exploitability and severity and penalised by conflict; A7 selects among seven graduated actions, each with a time-to-live and a scope restriction; and A8 observes outcomes, updates the policy and re-estimates reliability. Enforcement is suspended and the decision escalated when K exceeds K_max, and withheld below a risk floor.

Adversary model

Four adversary classes are distinguished by access rather than intent, since access determines which surfaces are reachable; Table 1 establishes, for each threat catalogued below, the weakest adversary capable of realizing it.

Table 1.

Adversary classes, by access and capability

Class	Access	Knowledge	Capability	Threats enabled
C1 External	Network traffic only	Black box; design may be known from publication	Craft protocol-valid flows; control volume and timing	T1, T2, T4, T9
C2 Lateral / insider	Coordination bus or one agent host	Grey box; agent interfaces and bus topics	Emit forged evidence; delay, drop or replay messages	T3, T5, T6
C3 Supply chain	Model artefacts and the update path	White box; training data and reward channel	Poison retraining batches; distort outcome telemetry	T7, T8
C4 Privileged misuse	Enforcement plane	White box	Drive the policy against chosen targets	T10

Attack surfaces and threat catalogue

Four surfaces are distinguished by the object that crosses them: the evidence path carries mass functions, the coordination bus messages, the learning path supervision, and the enforcement plane actions. Figure 1 locates the agents on these surfaces and marks where each threat attaches.

(cc) BY

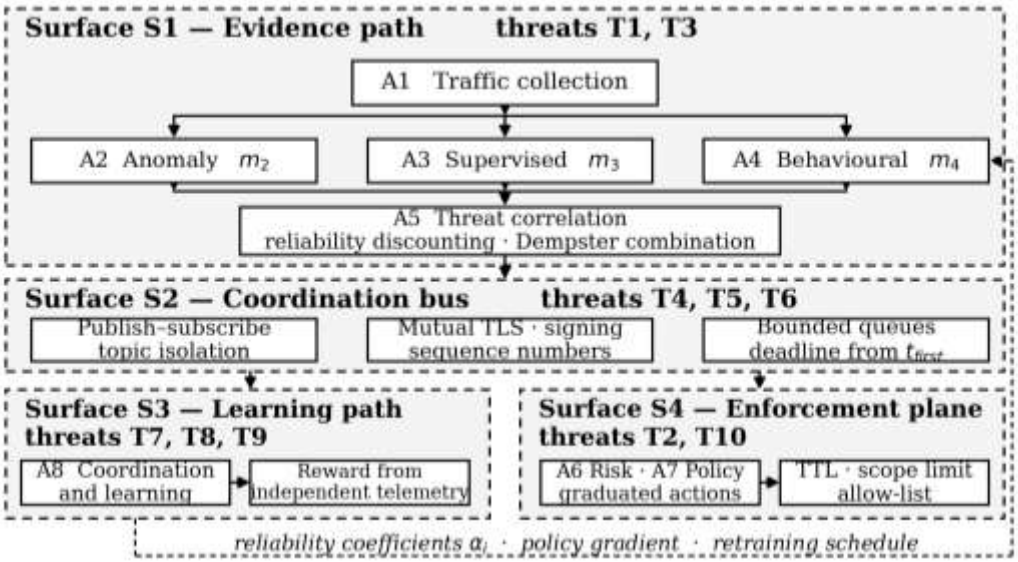


Figure 1. Attack surfaces of the eight-agent architecture and the threats that attach to each

Enumeration proceeded surface by surface, asking of each object crossing a surface what an adversary gains by forging, delaying, suppressing or observing it. Two entries of Table 2 constrained the architecture most: update-path poisoning and agent compromise together forbid the system to learn from its own unverified decisions, which is why A8 derives supervision from telemetry independent of A7's action history and from analyst adjudication.

Table 2.

Threat catalogue: control mapping and residual risk

ID	Threat	Surface	Architectural control	Residual risk
T1	Adversarial evasion	S1	Heterogeneous decision surfaces; disagreement raises conflict K	Evasion correlated across detector families still passes; 37.85 % remain silent misses
T2	Escalation saturation	S4	Risk-ordered escalation queue	Deliberately induced conflict saturates a 2× queue at 13.7 % adversarial volume
T3	Agent compromise	S1	Reliability discounting with α bounded to [0.30, 0.95]	Half of committed mass retained for 34 decisions; 31.6 % retained indefinitely
T4	Alert flooding and agent denial of service	S2	Bounded queues with deterministic shedding; deadline-bounded degradation	Degraded decisions reach 4.38 % at saturation; shed volume is reported, not recovered
T5	Message tampering and replay	S2	Signed messages with monotonic sequence numbers	Bounded by key custody; no protection once a signing key is captured with its agent
T6	Impersonation	S2	Mutual TLS with per-agent certificates	Same custody bound; revocation latency is not measured here
T7	Update-path poisoning	S3	Learning restricted to adjudicated labels and screened offline batches	Adjudication throughput bounds the retraining rate; analyst error propagates unchecked

(cc) BY

T8	Reward manipulation	S3	Reward derived from independent telemetry; updates frozen on anomalous divergence	The divergence threshold is itself a tunable beneath which an adversary may remain
T9	Model extraction	S3	Rate-limited observable response; graduated actions leak less than a binary block	Leakage reduced, not removed; time-to-live expiry is itself an observable signal
T10	Enforcement abuse	S4	Mandatory TTL, narrowest-scope invariant, allow-listed infrastructure, irreversible actions excluded	Bounded in time and scope, yet repeated short-lived actions still aggregate to disruption

Security-relevant parameters

Several parameters introduced for detection quality also determine security properties, and in two cases the setting serving one purpose weakens the other. The bounds $\alpha_{min} = 0.30$ and $\alpha_{max} = 0.95$ prevent, respectively, permanent exclusion of a recovered agent and the emergence of an individually decisive one; a decay of 0.98 smooths reliability against transient error; $K_{max} = 0.55$ suspends enforcement on contradictory evidence; a 40 ms deadline bounds the stall a slow agent can impose; and lifetimes of 180 to 900 s bound an incorrect enforcement. Sections 4.2 and 4.3 show the first and third carry security costs invisible from the detection objective that motivated them.

Evaluation setup

Measurements were taken on an Acer Nitro V 16S AI (ANV16S-41-R9SV) with an AMD Ryzen 7 260 processor, an RTX 5050 laptop graphics unit and 16 GB of DDR5-5600 memory, under Ubuntu 24.04.3 LTS with Python 3.13.5, PyTorch 2.9.0 and Stable-Baselines3 2.7.0; enforcement was exercised in Mininet 2.3.1b4 under an ONOS 2.7.0 controller. Models were trained on the corrected CIC-IDS2017 [1], [4] under a temporal split, with UNSW-NB15 [3] and CIC-DDoS2019 [2] as secondary corpora, and all figures are means over ten seeds. Scenario S6 applies benign traffic with induced legitimate anomalies and fixes the baseline escalation rate; S8 applies constrained perturbations, each flow remaining well formed for its protocol. The expected signature of a successful single-detector evasion here is a rise in conflict rather than a silent miss, and S8 tests that expectation.

Results

Behaviour under constrained evasion

Table 3 reports the effect of constrained perturbation on detection and on the decision route. Recall falls for every method, as it must; what distinguishes the architectures is where the lost instances go.

Table 3.

Behaviour under constrained protocol-valid perturbation (Scenario S8)

Quantity	Clean traffic	Perturbed traffic
Recall, supervised baseline (LightGBM)	0.8022	0.3874
Recall, proposed framework	0.8571	0.6215
Mean inter-detector conflict, K	0.147	0.612
Decisions exceeding $K_{max} = 0.55$ (%)	5.7	47.3
Attacks passing as silent misses, baseline (%)	19.78	61.26
Attacks passing as silent misses, framework (%)	14.29	37.85

The clean-traffic escalation figure of 5.7 % is measured under Scenario S6 and serves as the operational baseline. Read on detection alone the framework merely loses less: 23.56 points against 41.48. Read on the decision route the difference is qualitative, since mean conflict rises more than

(cc) BY

fourfold and nearly half of all evasion attempts cross the threshold and are escalated instead of enforced. Figure 2 decomposes the outcome, counting an escalated detection as flagged but not acted upon.

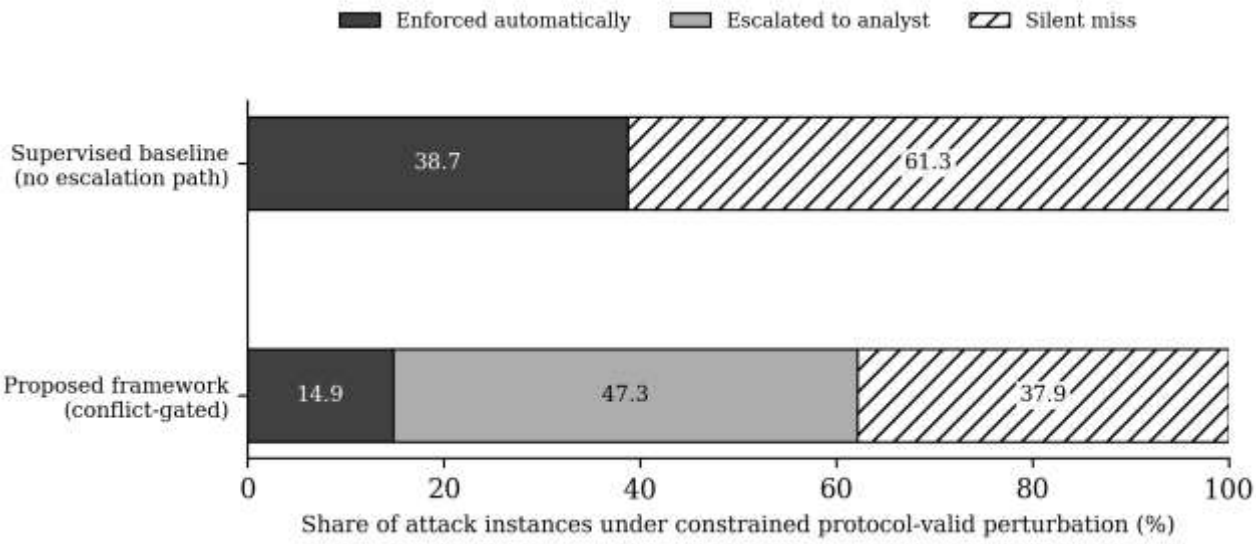


Figure 2. Composition of outcomes under constrained perturbation

The share of attacks passing silently falls from 61.26 % to 37.85 %, and roughly three-quarters of the framework's detections arrive as escalations rather than enforcement. The architecture does not defeat evasion; it converts much of it from an unobserved failure into an observed one. That is what a fusion rule returning a point estimate cannot supply, since a detector confident of attack opposed by one confident of benign is then indistinguishable from two that are merely uncertain.

Exposure window under agent compromise

Reliability discounting is the control assigned to agent compromise, and its adequacy is a question about dynamics rather than about the mechanism. The coefficient is an exponentially weighted agreement rate with decay 0.98, initialized at 0.95 and floored at 0.30, so an agent reporting adversarial has its committed mass scaled by 0.95 × 0.98^t after t adjudicated decisions until the floor is reached (Figure 3).

Two observations follow, the second unanticipated by the design. The exposure window is finite but long: half the compromised agent's committed mass survives 34 adjudicated decisions, and 57 are needed to reach the floor. More consequentially, the floor that keeps a transiently faulty agent from permanent exclusion simultaneously guarantees a compromised agent 31.6 % of its original influence for as long as it remains in service.

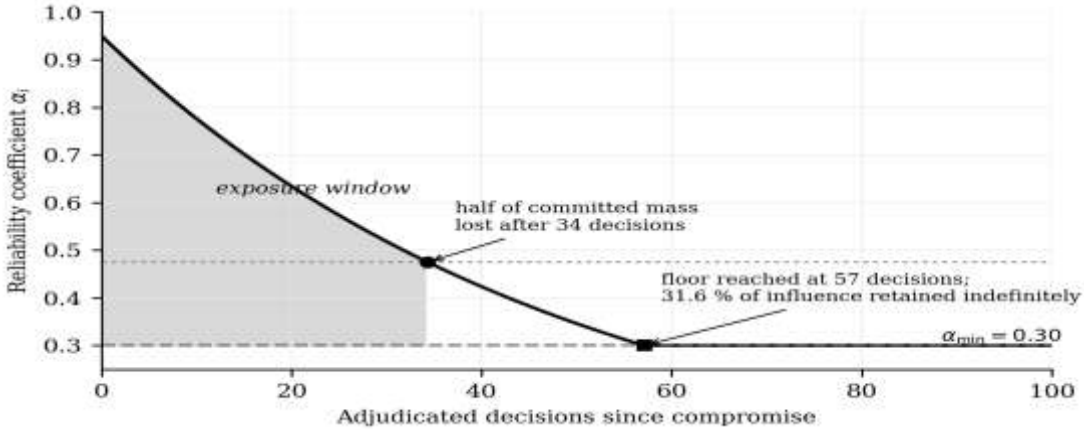


Figure 3. Reliability trajectory of a compromised agent

(cc) BY dtai.uz

Discounting bounds compromise asymptotically rather than removing it, and no Byzantine-robustness guarantee follows.

Escalation saturation

Section 4.1 established that the conflict gate converts evasion into escalation. That property is itself an attack surface: an adversary who understands the design need not evade the detectors, since inducing conflict suffices and consumes analyst rather than network capacity. Writing f for the adversarial share of decision volume and taking the measured rates of 5.7 % on ordinary and 47.3 % on perturbed traffic, the aggregate escalation rate is $E(f) = 0.057 + 0.416\,f$, evaluated in Table 4 and Figure 4 for several provisioning assumptions.

Table 4.

Adversarial volume required to saturate an escalation queue, by provisioned headroom

Provisioned headroom over baseline	Escalation capacity (% of decisions)	Adversarial volume at saturation (%)
1.25×	7.13	3.4
1.5×	8.55	6.9
2.0×	11.40	13.7
3.0×	17.10	27.4
5.0×	28.50	54.8
8.3×	47.31	100.0

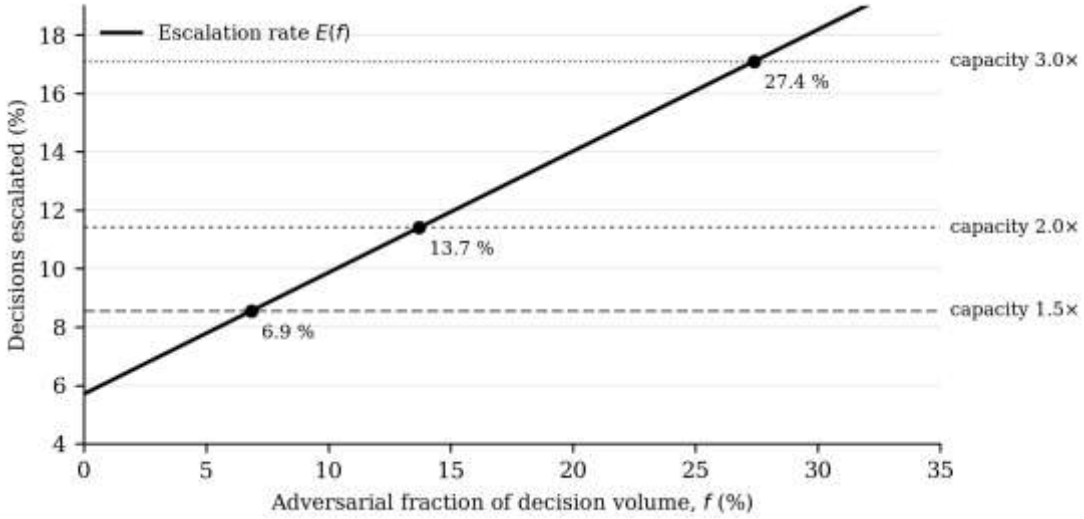


Figure 4. Escalation load against adversarial share of decision volume, with three provisioning levels

A queue sized at twice the baseline - a defensible choice absent this analysis - saturates once 13.7 % of decision volume is adversarial, and one sized at 1.5× at 6.9 %. Saturation degrades adjudication rather than detection, so escalated decisions accumulate unreviewed while enforcement continues on the remainder - the base-rate pressure identified by Axelsson [28] transferred from the alert stream to the queue. Three factors bound the attack, each with a limit: inducing conflict requires traffic one detector family confidently accepts while another rejects, harder than evading a single model but not impossible; escalations are risk-ordered, so high-consequence alerts are not displaced by volume, though the queue is finite; and sustained anomalous escalation is detectable, though detection does not restore capacity.

Coverage of the catalogue

Of the ten threats, three are bounded by cryptographic or structural means whose limits lie outside the architecture (T5, T6, T10), three quantitatively by mechanisms measured above (T1, T3, T4), two by procedural restrictions costing analyst throughput (T7, T8), one is reduced but not removed (T9), and one is created by the mitigation of another (T2). No threat is eliminated. The value of the catalogue is that every residual risk in Table 2 is a quantity an operator can provision against rather than an assurance that cannot be checked.

(cc) BY

Acknowledgement

This article was published within the framework of the project AL-9224104800 “Development of mechanisms for detecting and protecting network attacks in information and communication systems based on intelligent methods”.

Conclusion

An autonomous defence system is a target, and analysing it as one yields conclusions an evaluation of detection quality does not reach. Ten threats were enumerated across four surfaces of an eight-agent detect-and-respond architecture, each mapped to a control and a stated residual risk. Under constrained perturbation the architecture did not resist evasion so much as relocate its consequence, reducing the share of attacks passing silently from 61.26 % to 37.85 % while raising mean conflict from 0.147 to 0.612. Reliability discounting was shown to bound agent compromise only asymptotically, 31.6 % of a compromised agent’s influence surviving indefinitely by construction of the floor protecting recovered ones. And the conflict gate introduces escalation saturation, an availability attack on analyst capacity succeeding once 13.7 % of decision volume is adversarial. No threat is eliminated by its control; the contribution is that each residual risk is a quantity an operator can provision against, and that one of the architecture’s own safety mechanisms is the source of a threat it was not designed to create.

References

1. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP), Funchal, Portugal, 2018, pp. 108–116, doi: 10.5220/0006639801080116.
2. I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, “Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy,” in Proc. 2019 Int. Carnahan Conf. Security Technology (ICCST), Chennai, India, 2019, pp. 1–8, doi: 10.1109/CCST.2019.8888419.
3. N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems,” in Proc. 2015 Military Communications and Information Systems Conf. (MilCIS), Canberra, Australia, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.
4. G. Engelen, V. Rimmer, and W. Joosen, “Troubleshooting an intrusion detection dataset: The CICIDS2017 case study,” in Proc. 2021 IEEE Security and Privacy Workshops (SPW), 2021, pp. 7–12, doi: 10.1109/SPW53761.2021.00009.
5. L. Liu, G. Engelen, T. Lynar, D. Essam, and W. Joosen, “Error prevalence in NIDS datasets: A case study on CIC-IDS-2017 and CSE-CIC-IDS-2018,” in Proc. 2022 IEEE Conf. Communications and Network Security (CNS), 2022, pp. 254–262, doi: 10.1109/CNS56114.2022.9947235.
6. R. Flood, G. Engelen, D. Aspinall, and L. Desmet, “Bad design smells in benchmark NIDS datasets,” in Proc. 2024 IEEE 9th European Symp. Security and Privacy (EuroS&P), 2024, pp. 658–675.
7. D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, “Dos and don’ts of machine learning in computer security,” in Proc. 31st USENIX Security Symp., Boston, MA, USA, 2022, pp. 3971–3988.
8. G. Apruzzese, P. Laskov, E. Montes de Oca, W. Mallouli, L. Búrdalo Rapa, A. V. Grammatopoulos, and F. Di Franco, “The role of machine learning in cybersecurity,” Digital Threats: Research and Practice, vol. 4, no. 1, art. 8, Mar. 2023, doi: 10.1145/3545574.
9. S. Jacobs, R. Beltiukov, W. Willinger, R. A. Ferreira, A. Gupta, and L. Z. Granville, “AI/ML for network security: The emperor has no clothes,” in Proc. 2022 ACM SIGSAC Conf. Computer and Communications Security (CCS), 2022, pp. 1537–1551.
10. Z. Long and W. Jinsong, “A hybrid method of entropy and SSAE-SVM based DDoS detection and mitigation mechanism in SDN,” Computers & Security, vol. 115, art. 102604, 2022, doi: 10.1016/j.cose.2022.102604.
11. K. He, D. D. Kim, and M. R. Asghar, “Adversarial machine learning for network intrusion detection systems: A

(cc) BY

comprehensive survey," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 538–566, 2023, doi: 10.1109/COMST.2022.3233793.
12. T. T. Nguyen and V. J. Reddi, "Deep reinforcement learning for cyber security," IEEE Trans. Neural Networks and Learning Systems, vol. 34, no. 8, pp. 3779–3795, Aug. 2023, doi: 10.1109/TNNLS.2021.3121870.
13. K. Sethi, E. Sai Rupesh, R. Kumar, P. Bera, and Y. Venu Madhav, "A context-aware robust intrusion detection system: A reinforcement learning-based approach," International Journal of Information Security, vol. 19, pp. 657–678, 2020.
14. K. Sethi, Y. V. Madhav, R. Kumar, and P. Bera, "Attention based multi-agent intrusion detection systems using reinforcement learning," Journal of Information Security and Applications, vol. 61, art. 102923, 2021.
15. M. Tellache, A. Mokhtari, A. Amara Korba, and Y. Ghamri-Doudane, "Multi-agent reinforcement learning-based network intrusion detection system," in Proc. NOMS 2024 IEEE Network Operations and Management Symp., 2024, pp. 1–9.
16. G. Shi and G. He, "Collaborative multi-agent reinforcement learning for intrusion detection," in Proc. 2021 7th IEEE Int. Conf. Network Intelligence and Digital Content (IC-NIDC), 2021, pp. 245–249.
17. F. Louati, F. Barika Ktata, and I. Amous, "BiG-IDS: A decentralized multi-agent reinforcement learning approach for distributed intrusion detection in big data networks," Cluster Computing, 2024.
18. M. Foley, C. Hicks, K. Highnam, and V. Mavroudis, "Autonomous network defence using reinforcement learning," in Proc. 2022 ACM Asia Conf. Computer and Communications Security (ASIA CCS), 2022, pp. 1252–1254, doi: 10.1145/3488932.3527286.
19. H. Kheddar, D. W. Dawoud, A. I. Awad, et al., "Reinforcement-learning-based intrusion detection in communication networks: A review," IEEE Communications Surveys & Tutorials, 2024.
20. M. Standen, M. Lucas, D. Bowman, T. J. Richer, J. Kim, and D. Marriott, "CybORG: A gym for the development of autonomous cyber agents," in IJCAI-21 1st Int. Workshop on Cyber Defense, 2021, arXiv:2108.09118.
21. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv:1707.06347, 2017, doi: 10.48550/arXiv.1707.06347.
22. M. K. Adawadkar and N. Kulkarni, "Cyber-security and reinforcement learning - A brief survey," Engineering Applications of Artificial Intelligence, vol. 114, art. 105116, 2022, doi: 10.1016/j.engappai.2022.105116.
23. H. Benaddi, K. Ibrahimi, A. Benslimane, M. Jouhari, and J. Qadir, "Robust enhancement of intrusion detection systems using deep reinforcement learning and stochastic game," IEEE Trans. Vehicular Technology, vol. 71, no. 10, pp. 11089–11102, 2022.
24. S. Bozorov, N. Nasrullayev, F. Xoliyorov, and T. Buriboyev, "Cooperative multi-agent intrusion detection and mitigation: Heterogeneous AI detectors with evidence-theoretic fusion and risk-aware reinforcement-learned response," submitted for publication, 2026.
25. A. Servin and D. Kudenko, "Multi-agent reinforcement learning for intrusion detection," in Adaptive Agents and Multi-Agent Systems III, Berlin, Germany: Springer, 2008, pp. 211–223.
26. G. Folino and P. Sabatino, "Ensemble based collaborative and distributed intrusion detection systems: A survey," Journal of Network and Computer Applications, vol. 66, pp. 1–16, 2016, doi: 10.1016/j.jnca.2016.03.011.
27. D. Jin, S. Chen, H. He, X. Jiang, S. Cheng, and J. Yang, "Federated incremental learning based evolvable intrusion detection system for zero-day attacks," IEEE Network, vol. 37, no. 1, pp. 125–132, 2023.
28. . Axelsson, "The base-rate fallacy and its implications for the difficulty of intrusion detection," in Proc. 6th ACM Conf. Computer and Communications Security (CCS), Singapore, 1999, pp. 1–7, doi: 10.1145/319709.319710.

(cc) BY